15

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/827,700	04/06/2001	James R. Lewis	6169-188	5073

7590 11/26/2003

Gregory A. Nelson
Akerman Senterfitt
222 Lakeview Avenue, Fourth Floor
P.O. Box 3188
West Palm Beach, FL 33402-3188

EXAMINER	
BRANT, DMITRY	
ART UNIT	PAPER NUMBER
2655	2

DATE MAILED: 11/26/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

TS

Office Action Summary	**Application No.**	**Applicant(s)**	
	09/827,700	LEWIS ET AL.	
	Examiner	**Art Unit**	
	Dmitry Brant	2655	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *5/10/2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *5/10/2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .

4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-23 are rejected under 35 U.S.C. 103(a) as being obvious over Yushik (6,526,382) in view of VoiceXML specification (published 17/8/1999, located at www.voicexml.org/specs/VoiceXML-0.9-19990817.pdf)

3. As per claim 1, Yushik discloses a system comprised of Voice Activated User Interface (10, FIG. 1) through speaker and microphone (12, 14, FIG. 1), utilizing two alternate prompting structures (Column 4, lines 1-4) depending on user errors (Col. 4, lines 7-9). In addition, Yushik's system is implemented in software (Column 3, lines 17-19).

Yushik does not disclose "said prompt delivery system delivering tapered prompts in response to the determination of a quantity of said at least one of correct and incorrect responses."

VoiceXML teaches the use of "tapered" prompts, where each prompt has an associated "count" attribute. (13.6, Page 36). Each form field and each menu has an internal prompt counter that is reset to one each time the form or menu is entered. Whenever the system uses a prompt, its associated prompt counter is incremented. Additionally, VoiceXML has control flow mechanisms with “if” and “goto” statements (26, 27, Page 46) that allow programmers to conditionally switch between multiple forms. Hence, one could construct two sets of forms that use full and tapered prompts. The system would use the full prompt forms after the number of mistakes exceeds the value of the associated prompt counter. Alternatively, one could construct a form or a menu where a “tapered prompt” is activated after the number of correct answers exceeds the counter value. (13.6, page 36)

At the time of the invention it would have been obvious to a person of ordinary skill in the art that the system taught by Yushik could be modified to use the VoiceXML software to count the correct or incorrect answers and switch to the corresponding prompt depending the on the count value. This would allow Yushik’s system to easily implement the said “tapered prompts” in the industry-accepted programming language and take advantage of the flexibility and rich functionality of VoiceXML.

4. As per claim 2, Yushik does not disclose a system ”wherein said prompt delivery system delivers tapered prompts in response to correct responses to a prerequisite number of full prompts.”

However, VoiceXML teaches the use of prompts with associated counters (13.6, Page 36). Hence, once could write a form that invokes a shorter, “tapered” prompt only after a prerequisite number of full prompts (<prompt count = “prereq. number”>) have been answered correctly, as illustrated in example on page 36. Therefore, at the time of the invention it would have been obvious to a person of ordinary skill in the art that the system taught by Yushik could be modified to use the VoiceXML software to invoke "tapered" prompts only after N full prompts have been correctly answered, because this would allow the user who proved that he is sufficiently accustomed to the system to proceed to quicker version of prompts.

5. As per claim 3, Yushik does not disclose "said prompt delivery system delivers tapered prompts to said user until said user provides an incorrect response."

However, VoiceXML teaches a way to switch between forms using "if and "goto" statements (26,27 Page 46). It would have been obvious to a person of ordinary skill in the art that one can write a program flow in VoiceXML that would switch to a different type of form (in this case full-length) in response to an incorrect answer and use that program with the system taught by Yushik, because this would make sure that the system delivers more information if user starts making mistakes while using shorter prompts.

6. As per claim 4, Yushik does not disclose a system "wherein said prompt delivery system delivers full prompts following an incorrect response until a subsequent prerequisite number of correct responses are provided by a user."

VoiceXML teaches the use of prompt counters (13.6, Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can use an associated prompt counter in the "full" form to check whether a prerequisite number of correct responses have been provided by the user, and if so, then switch to the "tapered" version of prompt using "goto" statement, because this configuration would allow the user to switch to quicker prompts once he proves a sufficient knowledge of the prompting system.

7. As per claim 5, Yushik does not disclose a system "wherein said prerequisite number of correct responses to full prompts is increased if an incorrect response is received."

VoiceXML teaches the use of variables (9, Page 25) along with the use of prompt counters (13.6, Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that stores the number of prerequisite correct responses in a variable declared within a form and increments that variable when an incorrect response is received and then uses value of this variable as a prerequisite number of correct responses. This decreases the chance of the system switching back to tapered prompts before the user is fully ready to start using tapered prompts and insures that the system increases the switching requirements for the user as he makes more and more mistakes.

8. As per claim 6, Yushik does not disclose a system "wherein said prerequisite number of responses to full prompts is increased for each sequential incorrect response that is received."

VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that stores the number of prerequisite correct responses in a variable declared within a form and increments that variable for each sequential incorrect response. This decreases the chance of the system switching back to tapered prompts before the user is fully ready to start using tapered prompts and insures that the system increases the switching requirements for the user as he makes more and more mistakes, particularly because sequential mistakes indicate that the user is not making mistakes by accident and is actually having difficulty with the system.

9. As per claim 7, Yushik does not disclose a system "wherein said incorrect responses include at least one selected from the group consisting of out-of-grammar responses, silence time outs, and help responses."

VoiceXML teaches prompts that support the handling of incorrect grammar, timeouts , and help requests (13.6, Page 36 and 13.7, Page 37). It would have been obvious to a person of ordinary skill in the art to use VoiceXML with Yushik's system because this would allow for the support of different types of indicators of user's difficulty with the system: incorrect responses, pauses and calls for help.

10. As per claim 8, Yushik does not disclose a system "wherein said prompt delivery system delivers tapered prompts in response to a determination that correct responses have been provided by a user to a minimum prerequisite proportion of said prompts."

However, VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) stores the number of correct and incorrect responses in two variables declared within a form, (2) calculates the percentage of the correct responses to the total number of responses and (3) switches to the "tapered" prompt using "goto" statement once a prerequisite proportion of the correct responses has been reached. This would allow the system to switch to "tapered" mode if user proved that he was "on average" capable of using the tapered prompts and was correct in, say, 90% of the times.

11. As per claim 9, Yushik discloses the use of segmented interfaces using "chunks" (Column 18, lines 33-36).

Yushik does not disclose "a prompt delivery system [that] delivers tapered prompts in a segment in response to a determination that correct responses have been provided by a user to a prerequisite quantity of said prompts while in said segment."

VoiceXML teaches the use of prompt counters (13.6, Page 36). In addition, one could construct forms using VoiceXML grammar to segment the overall user interface.

It would have been obvious to a person of ordinary skill in the art to combine the segmented interface disclosed by Yushik with the prompt-counting ability of VoiceXML to switch to "tapered" prompt mode using VoiceXML flow control mechanisms when a user provides a prerequisite quantity of correct responses. This would allow the user to navigate through the menu segments using the "tapered" prompts when the user proves to the system that he is sufficiently proficient with the use of the system.

12. As per claim 10, Yushik does not disclose a system "wherein said prerequisite quantity of correct responses comprises a prerequisite number of sequential correct responses provided by the user while in said segment."

However, VoiceXML teaches the use of variables (9, Page 25) along with the use of prompt counters (13.6, Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) sequentially increments the variable counter in the form that corresponds to a said segment and (2) switches to the 'tapered" format once the number of correct sequential answers has been reached. This would prove to the system that the user is capable of using "tapered" prompts because he keeps answering the questions correctly.

13. As per claim 11, Yushik does not disclose a system "wherein said prerequisite quantity of correct responses comprises a prerequisite minimum proportion of correct responses provided by the user while in said segment."

VoiceXML VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) stores the number of correct and incorrect responses in two variables declared within a form, (2) calculates the percentage of the correct responses to the total number of responses and (3) switches to the "tapered" prompt within a segment using "goto" statement once a prerequisite proportion of the correct responses has been reached. This would allow the system to switch to "tapered" mode if user proved that he was "on average" capable of using the tapered prompts and was correct in, say, 90% of the times.

14. Claim 12 is rejected under 35 U.S.C. 103(a) as being obvious over Yushik in view of VoiceXML specification.

As per claim 12, Yushik discloses a system comprised of Voice Activated User Interface (10, FIG. 1) through speaker and microphone (12, 14, FIG. 1), utilizing two alternate prompting structures (Column 4, lines 1-4) depending on user skill level. In addition, Yushik's system is implemented in software (Column 3, lines 17-19).

Yushik does not disclose "determining the quantity of at least one of correct and incorrect responses to said user prompts, said prompt delivery system delivering tapered prompts in response to the determination of a quantity of said at least one of correct and incorrect responses."

VoiceXML teaches the use of "tapered prompts", where each prompt has an associated "count" attribute. (13.6, Page 36). Each form field and each menu has an internal prompt counter that is reset to one each time the form or menu is entered. Whenever the system uses a prompt, its associated prompt counter is incremented. Additionally, VoiceXML has control flow mechanisms with "if" and "goto" statements (26,27 Page 46) that allow programmers to conditionally switch between multiple forms. Hence, one could construct two sets of forms that use full and tapered prompts. The system would use the full prompt forms after a number of mistakes exceed the value of the associated prompt counter. Alternatively, one could construct a form or a menu where a "tapered prompt" is activated after the number of correct answers exceeds the counter value. (See example, 13.6, page 36)

At the time of the invention it would have been obvious to a person of ordinary skill in the art that the system taught by Yushik could be modified to use the VoiceXML software to count the correct or incorrect answers and switch to the corresponding prompt depending the on the count value. This would allow Yushik's system to easily implement the said "tapered prompts" in the industry-accepted programming language and take advantage of the flexibility and rich functionality of VoiceXML.

15. As per claim 13, Yushik does not disclose a system " wherein tapered prompts are delivered in response to determining that correct responses have been provided by a user to a prerequisite number of full prompts."

However, VoiceXML teaches the use of prompts with associated counters (13.6, Page 36). Hence, once could write a form that invokes a shorter, "tapered" prompt only after a prerequisite number of full prompts (<prompt count = "prereq. number">) have been answered correctly, as illustrated in example on page 36. Therefore, at the time of the invention it would have been obvious to a person of ordinary skill in the art that the system taught by Yushik could be modified to use the VoiceXML software to invoke "tapered" prompts only after a prerequisite number of full prompts have been correctly answered, because this configuration would allow the user to switch to quicker prompts once he proves the sufficient knowledge of the prompting system.

16. As per claim 14, Yushik does not disclose a system "wherein tapered prompts are delivered when a prerequisite number of correct responses to full prompts are received."
However, VoiceXML teaches the use of prompts with associated counters (13.6, Page 36). Hence, once could write a form that invokes a shorter, "tapered" prompt only after a prerequisite of correct responses have been received. Therefore, at the time of the invention it would have been obvious to a person of ordinary skill in the art that the system taught by Yushik could be modified to use the VoiceXML software to invoke "tapered" prompts only after a prerequisite number of correct responses have been received, because this configuration would allow the user to switch to quicker prompts once he proves the sufficient knowledge of the prompting system.

17. As per claim 15, Yushik does not disclose a system "wherein tapered prompts are delivered to said user until said user provides an incorrect response."

However, VoiceXML teaches a way to switch between forms using "if and "goto" statements (26,27 Page 46). It would have been obvious to a person of ordinary skill in the art that one can write a program flow in VoiceXML that would switch to a different type of form (in this case full-length) in response to an incorrect answer, because this would make sure that the system delivers more information if user starts making mistakes while using shorter prompts.

18. As per claim 16, Yushik does not disclose a system "wherein full prompts are delivered following an incorrect response until a prerequisite number of correct responses are provided by a user."

VoiceXML teaches the use of prompt counters (13.6, Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can use an associated prompt counter in the "full" form to check whether a prerequisite number of correct responses have been provided by the user, and if so, then switch to the "tapered" version of prompt using "goto" statement, because this configuration would allow the user to switch to quicker prompts once he proves a sufficient knowledge of the prompting system.

19. As per claim 17, Yushik does not disclose a system "wherein said prerequisite number of correct responses to full prompts is increased if an incorrect response is received."

VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that stores the number of prerequisite correct responses in a variable declared within a form and increments that variable when an incorrect response is received. This decreases the chance of the system switching back to tapered prompts before the user is fully ready to start using tapered prompts and insures that the system increases the switching requirements for the user as he makes more and more mistakes.

20. As per claim 18, Yushik does not disclose a system "wherein said prerequisite number of responses to full prompts is increased for each sequential incorrect response that is received."

VoiceXML teaches the use of variables (9, Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that stores the number of prerequisite correct responses in a variable declared within a form and increments that variable for each sequential incorrect response. This decreases the chance of the system switching back to tapered prompts before the user is fully ready to start using tapered prompts and insures that the system increases the switching requirements for the user as he makes more and more mistakes, particularly because sequential mistakes indicate that the user is not making mistakes by accident and is actually having difficulty with the system.

21. As per claim 19, Yushik does not disclose a system "wherein said incorrect responses include at least one selected from the group consisting of out-of-grammar responses, silence time outs, and help responses."

VoiceXML teaches prompts that support the handling of incorrect grammar, timeouts , and help requests (13.6, Page 36 and 13.7, Page 37)). It would have been obvious to a person of ordinary skill in the art to use VoiceXML with Yushik's system because this would allow for the support of different types of indicators of user's difficulty with the system: incorrect responses, pauses and calls for help.

22. As per claim 20, Yushik does not disclose a system "wherein tapered prompts are delivered in response to determining that correct responses have been provided by a user to a minimum prerequisite proportion of said prompts."

However, VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) stores the number of correct and incorrect responses in two variables declared within a form, (2) calculates the percentage of the correct responses to the total number of responses and (3) switches to the "tapered" prompt using "goto" statement once a prerequisite proportion of the correct responses has been reached. This would allow the system to switch to "tapered" mode if user proved that he was "on average" capable of using the tapered prompts and was correct in, say, 90% of the times.

23. As per claim 21, Yushik discloses the use of segmented interfaces using "chunks" (Column 18, lines 33-36).

Yushik does not disclose a system where "tapered prompts are delivered in a segment in response to determining that correct responses have been provided by a user to a prerequisite quantity of said prompts while in said segment"

VoiceXML teaches the use of prompt counters (13.6, Page 36). In addition, one could construct forms using VoiceXML grammar to segment the overall user interface.

It would have been obvious to a person of ordinary skill in the art to combine the segmented interface disclosed by Yushik with the prompt-counting ability of VoiceXML to switch to "tapered" prompt mode using VoiceXML flow control mechanisms when a user provides a prerequisite quantity (such as number, proportion) of correct responses. This would allow the user to navigate through the menu segments using the "tapered" prompts when the user proves to the system that he is sufficiently proficient with the use of the system.

24. As per claim 22, Yushik does not disclose a system "wherein said prerequisite quantity of correct responses comprises a prerequisite number of sequential correct responses provided by the user while in said segment."

However, VoiceXML teaches the use of variables (9, Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) sequentially increments the variable counter in the form that corresponds to a said segment and (2) switches to the 'tapered" format once the number of correct sequential answers has been reached. This would prove to the system that the user is capable of using "tapered" prompts because he keeps answering the questions correctly.

25. As per claim 23, Yushik does not disclose a system "wherein said prerequisite quantity of correct responses comprises a prerequisite minimum proportion of correct responses provided by the user while in said segment."

VoiceXML VoiceXML teaches the use of variables (Page 25) along with the use of prompt counters (Page 36). It would have been obvious to a person of ordinary skill in the art that a programmer can write a form in VoiceXML that (1) stores the number of correct and incorrect responses in two variables declared within a form, (2) calculates the percentage of the correct responses to the total number of responses and (3) switches to the "tapered" prompt within a segment using "goto" statement once a prerequisite proportion of the correct responses has been reached. This would allow the system to switch to "tapered" mode if user proved that he was "on average" capable of using the tapered prompts and was correct in, say, 90% of the times.

Conclusion

26. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Parthasarthi (5,204,968) discloses a system that changes user prompts based on the quality of user responses.

Eisen et al. (4,964,077) discloses a system that adjusts help information based on the user's familiarity with the system.

27. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dmitry Brant whose telephone number is (703) 305-8954. The examiner can normally be reached on Mon. - Fri. (8:30am - 5pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Talivaldis Ivars Smits can be reached on (703) 306-3011. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 306-0377.

DB
11/17/03

TĀLIVALDIS IVARS ŠMITS
PRIMARY EXAMINER